CLAIMS

I claim:

1. A method for scheduling a plurality of primary and secondary transactions between a first device and a second device across a shared bus, comprising the steps of:

(a) splitting an allocated bus time for the shared bus into frames of equal length;

(b) receiving a bus request and determining whether the bus request in a current frame is for a primary transaction or a secondary transaction;

(c) processing a primary transaction bus request if one exists, otherwise processing a secondary transaction bus request if one exists, wherein secondary transaction bus requests are queued if received while one of the primary transaction bus requests is currently being processed;

(d) repeating steps (b) through (c) until a current frame time has ended; and

(e) processing queued secondary transaction bus requests at the start of a new frame before continuing with step (b), wherein the new frame becomes the current frame of step (b).

2. The method of claim 1, wherein the primary transaction is an asynchronous transaction and the secondary transaction is an isochronous transaction.

3. The method of claim 1, wherein the secondary transaction bus requests of step (c) are processed in the order in which they were received, and wherein secondary transaction bus requests from an immediately prior frame are processed before secondary transaction bus requests of the current frame.

4. The method of claim 1, wherein the queued secondary transaction bus requests of step (e) are processed in a round-robin fashion at the start of the new frame.

5. The method of claim 1, wherein the shared bus is an Ethernet connection.

6. The method of claim 1, wherein the shared bus is a Universal Serial Bus.

7. The method of claim 1, wherein the shared bus is an IEEE 1394 bus.

8. The method of claim 1, wherein the shared bus is a PCI bus.

9. The method of claim 1, wherein the first and second devices are workstations.

10. The method of claim 1, wherein the first and second devices are peripherals.

11. The method of claim 1, wherein the first and second devices are Windows-based servers.

12. The method of claim 1, wherein the first device is a workstation and the second device is a Windows-based server.

13. A computer system having the capability of scheduling transactions across a shared bus, comprising:

a processor;

a mass storage device coupled to the processor;

transaction scheduling code stored in a processor readable medium for causing the processor to perform the steps of:

(a) splitting an allocated bus time for the shared bus into frames of equal length;

(b) receiving a bus request and determining whether the bus request in a current frame is for a primary transaction or a secondary transaction;

(c) processing a primary transaction bus request if one exists, otherwise processing a secondary transaction bus request if one exists, wherein secondary transaction bus requests are queued if received while one of the primary transaction bus requests is currently being processed;

(d) repeating steps (b) through (c) until a current frame time has ended; and

(e) processing queued secondary transaction bus requests at the start of a new frame before continuing with step (b), wherein the new frame becomes the current frame of step (b).

14. The computer system claim 13, wherein the primary transaction is an asynchronous transaction and the secondary transaction is an isochronous transaction.

15. The computer system of claim 13, wherein secondary transaction bus requests of step (c) are processed in the order in which they were received, and wherein secondary transaction bus requests from an immediately prior frame are processed before secondary transaction bus requests of the current frame.

16. The computer system of claim 13, wherein the queued secondary transaction bus requests of step (e) are processed in a round-robin fashion at the start of the new frame.

17. The computer system of claim 13, wherein the shared bus is an Ethernet connection.

18. The computer system of claim 13, wherein the shared bus is a Universal Serial Bus.

19. The computer system of claim 13, wherein the shared bus is an IEEE 1394 bus.

20. The computer system of claim 13, wherein the shared bus is a PCI bus.

21. The computer system of claim 13, wherein the processor is coupled to a workstation by the shared bus.

22. The computer system of claim 13, wherein the processor is coupled to a peripheral by the shared bus.

23. The computer system of claim 13, wherein the processor is coupled to a Windows-based server by the shared bus.

24. The computer system of claim 13, wherein the computer system is a workstation that is coupled to a Windows-based server by the shared bus.